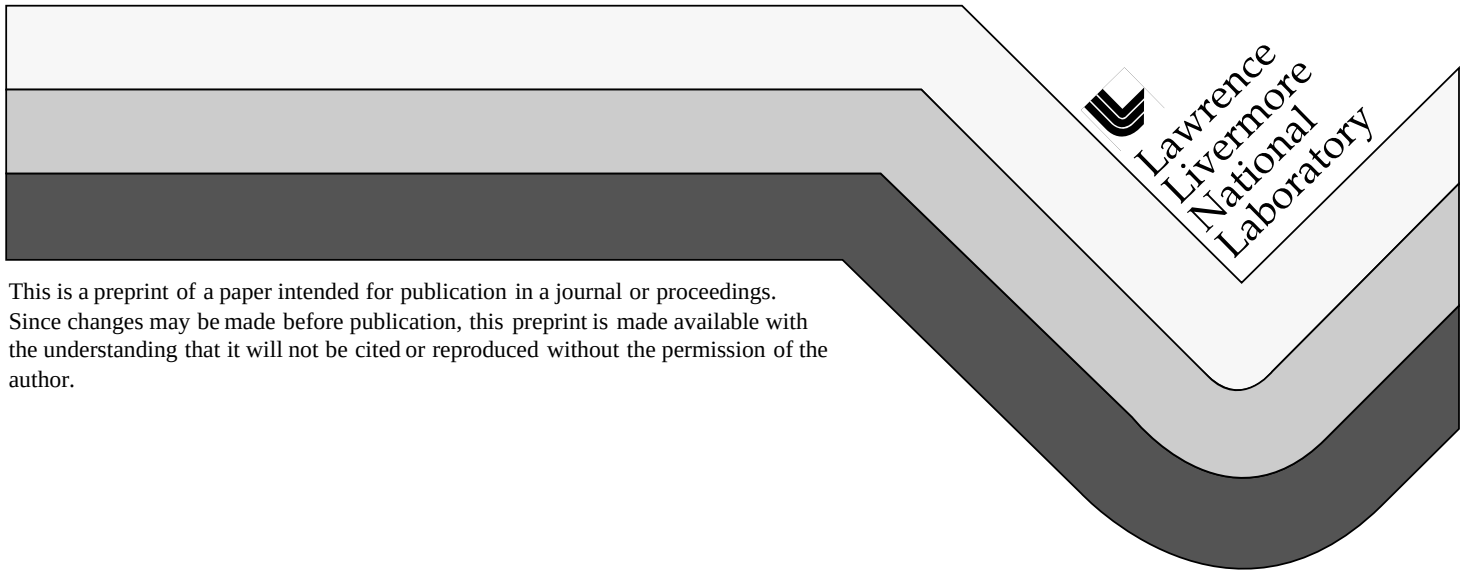
UCRL-JC-133095
PREPRINT

Domain Decomposition Methods for a Parallel Monte Carlo Transport Code

H. J. Alme, G. H. Rodrigue, G. B. Zimmerman

This paper was prepared for submittal to the
1998 Nuclear Explosives Development Conference
Las Vegas, NV
October 25-30, 1998

January 27, 1999



Domain Decomposition Methods for a Parallel Monte Carlo Transport Code[1]

Henry J. Alme, Garry H. Rodrigue, and George B. Zimmerman
Lawrence Livermore National Laboratory

Achieving parallelism in simulations that use Monte Carlo transport methods presents interesting challenges. For problems that require domain decomposition, load balance can be harder to achieve. The Monte Carlo transport package may have to operate with other packages that have different optimal domain decompositions for a given problem. To examine some of these issues, we have developed a code that simulates the interaction of a laser with biological tissue; it uses a Monte Carlo method to simulate the laser and a finite element model to simulate the conduction of the temperature field in the tissue. We will present speedup and load balance results obtained for a suite of problems decomposed using a few domain decomposition algorithms we have developed.

Keywords: parallel processing, Monte Carlo transport, domain decomposition

1 Introduction

Parallel processors have great potential in scientific computing; Realizing that potential has been difficult in many areas. Using parallel computers on problems involving Monte Carlo transport presents new issues. The locality of the calculation in Monte Carlo transport makes domain decomposition more challenging.

A typical domain decomposition method for a calculation involving a differencing scheme will be independent of the physics involved. Recursive Spectral Bisection [1]—a popular decomposition algorithm for problems in difference schemes—considers only the mesh in deciding the decomposition. For Monte Carlo transport calculations, a decomposition method that considers the physics of the problem may have a better chance at success.

This paper is laid out as follows: Section 2 briefly outlines the model used to simulate laser-tissue interaction. In Section 3, we discuss some of the issues involved in a parallel laser-tissue code. Section 4 discusses domain decomposition of Monte Carlo transport calculations and explains our approach. We present the results of some test problems in section 5. Section 6 combines a summary with a brief discussion of some possible future work.

2 Laser-Tissue Models

For this paper, we examine parallel processing of a problem in laser-tissue interaction. The model has two modules: a Monte Carlo transport module that models laser light scattering

[1]Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

and absorption using Monte Carlo photonics, and a finite element module that models the thermal heat conduction and tissue damage of the tissue under laser irradiation.

2.1 Photon Transport

The collision density $\phi(\vec{x},\vec{\Omega})$ satisfies the steady state transport equation:

$$\phi(\vec{x},\vec{\Omega}) = S_c(\vec{x},\vec{\Omega}) + \int\int K(\vec{x}',\vec{\Omega}' \rightarrow \vec{x},\vec{\Omega})\phi(\vec{x}',\vec{\Omega}')d\vec{x}'d\vec{\Omega}' \tag{1}$$

The solution is computed each time step using Monte Carlo photon transport. The photons are scattered using a Henyey-Greenstein phase function with an anisotropy factor $g = .9$, which has been found to be a reasonable model for biological tissue [2, 3].

Each photon in the simulation is a "super photon" representing many photons. Each photon starts with a statistical weight set to 1. As the photon moves through the computational grid, it's weight is exponentially attenuated using the absorption coefficient μ_a. We use Russian Roulette to decide when to terminate a particle history. When a particle's weight drops below a user-defined level, a random number determines at each collision if the particle continues or if its history is terminated and its remaining weight deposited in its current zone.

2.2 Tissue Response

2.2.1 Thermal Transfer.

The energy balance for tissue under laser irradiation is

$$\rho c_v \frac{\partial}{\partial t} T(\vec{x},t) = -\nabla \cdot \vec{q} + S(\vec{x},t) \tag{2}$$

where ρ is the density of the material, c_v is the specific heat, $\vec{q}$ is the conductive heat flux vector, and S is a source term consisting of separate terms reflecting, for example, energy added due to laser irradiation or lost due to cooling from blood circulation. For a homogeneous isotropic medium, the conductive heat flux vector is given by the Fourier Law as

$$\vec{q} = -\kappa \nabla T(\vec{x},t) \tag{3}$$

where κ is the thermal conductivity of the medium. $S_l(\vec{x},t)$, the source term due to the laser, is proportional to teh collisiosn density and is given by

$$S_l(\vec{x},t) = B\phi(\vec{x},t) \tag{4}$$

where $\phi(\vec{x},t)$ is the collision density of radiation with the material.

2.2.2 Tissue Damage.

Damage to the tissue can be modeled using a first order Arhennius rate equation[4]. Damage is represented by a parameter $\Omega(t)$ that is calculated for each zone by integrating the rate equation

$$\Omega(t) = \frac{k_b}{h} \int_0^t T(\tau) \exp\left[\frac{\Delta S}{R} - \frac{\Delta H}{RT(\tau)}\right] d\tau \tag{5}$$

where k_b, h, and R are Boltzmann's Constant, Planck's Constant, and the gas constant, respectively. ΔH and ΔS are the enthalpy and entropy of the reaction, respectively.

The damage Ω affects the optical properties in a zone. The scattering coefficient μ_s varies between two prescribed values: the undamaged value $\mu_{s,u}$ and the fully damaged value $\mu_{s,d}$ according to the relation

$$\mu_s(t) = \mu_{s,u} e^{-\Omega(t)} + \mu_{s,d}(1 - e^{-\Omega(t)}) \tag{6}$$

Thus the scattering coefficient in a zone will begin at $\mu_{s,u}$ when $\Omega(0) = 0$ and will move monotonically toward $\mu_{s,d}$ as Ω increases.

3 Parallelizing Laser-Tissue Calculations

3.1 Domain Decomposition for Finite Elements

Domain decomposition for a finite element calculation is reasonably well understood. There are well-known algorithms, such as Recursive Spectral Bisection [1], that will provide reasonable decompositions of the computational mesh.

There are two rules of thumb for decomposing finite element or finite difference problems. The first rule is that the decomposition should assign approximately equal numbers of zones to each processor – this ensures load balance. The second rule concerns communications costs; the ratio of computing to communication time heavily influences the speedup of a calculation. This ratio is in turn related to the surface area to volume ratio of subdomains.

For a finite element calculation, RSB will usually provide a decomposition that takes the above into account. It will generally provide similar sized domains with a minimal amount of communication required.

3.2 Parallel Monte Carlo

The previous rationale for using the RSB algorithm does not apply to a Monte Carlo transport calculation. There are two issues in parallelizing Monte Carlo calculations absent in finite element calculations.

First, the computational effort is not evenly distributed over the mesh. Some zones in a Monte Carlo transport calculation will have more particle collisions, requiring more computation time. Simply getting the same number of zones on each processor will not necessarily guarantee load balance.

The second issue is the lack of *a priori* knowledge of the work distribution for a given calculation. The stochastic nature of Monte Carlo transport calculations makes it impossible to predict exactly where the sample particles will go in the course of a calculation.

4 Decomposing Monte Carlo

4.1 Types of Parallelism for Monte Carlo

4.1.1 Task Farm Parallelism.

Task Farm parallelism (Fig. 1) takes advantage of the independence of the individual photon histories involved in a single Monte Carlo time step. Each processor has a complete copy of the computational mesh and tracks a fraction of the photon histories desired. The final answer comes from a global reduction after all photons have been tracked.

Task farm parallelism is the method used in past work on parallel Monte Carlo transport calculations (see, for example, [5, 6, 7, 8, 9]), due to its ease of use. Only two issues arise: the need for the individual processors to have independent random number streams, and the need for the global reduction at the end of each time step, both easily handled. The literature is extensive on parallel random number issues (see [10, 11] for examples), and the global reduction requires a minimum of new design and coding.

As problems continue to grow in resolution and as three dimensional problems become common, this approach becomes less useful. For some problems, the entire computational mesh will not fit into the memory of a single processor. This requires a different approach, as discussed below.

4.1.2 Spatial Parallelism.

Spatial parallelism (Fig. 2) is a more typical method in computational physics. Each processor gets a portion of the computational mesh, a subdomain. The particles then move about as normal, but a particle now may cross a boundary between processors. When this happens, the processor hands the particle off to the processor that owns the new domain. This requires message passing.

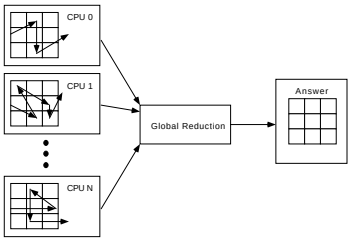


Figure 1: Task Farm Parallelism. Final answer via global reduction

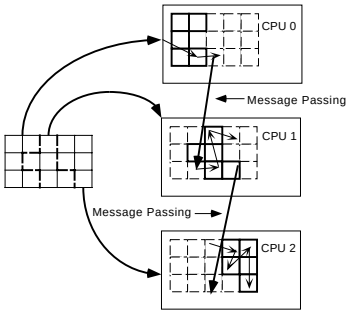


Figure 2: Spatial Parallelism. Particles cross CPU boundaries via messaging.

4.1.3 A Hybrid Approach

The parallelized Monte Carlo transport problems in this paper use a hybrid approach, combining task farm parallelism with spatial parallelism. The computational mesh is divided into subdomains small enough to fit onto a processor; there will typically be more processors available than subdomains. The subdomains will be replicated across some subset of the processors, ensuring that each subdomain is assigned to at least one processor (see Fig. 3).

This approach allows for some flexibility. Subdomains requiring more computational effort can be replicated on more processors than subdomains requiring less effort. This allows for the best load balance possible, maximizing the performance increase from the parallel computer. The issue then becomes one of determining the amount of effort needed in a subdomain.

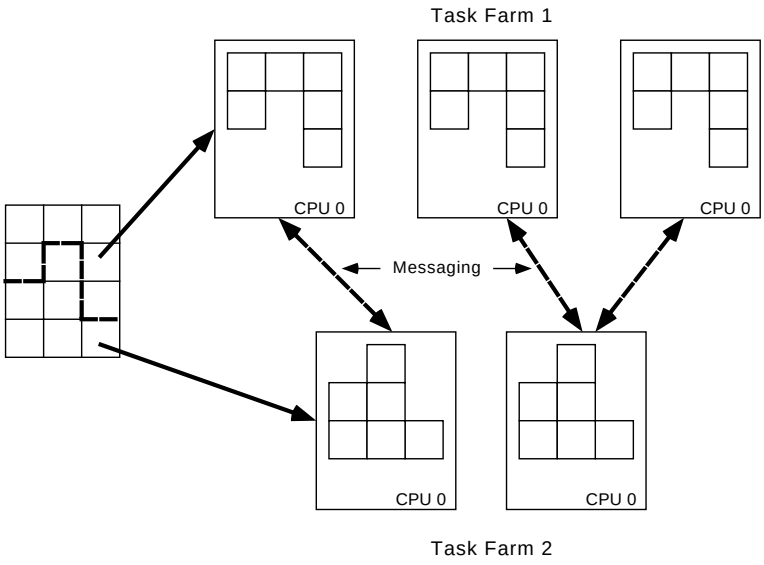


Figure 3: A Hybrid Approach. Combination of task farm and spatial parallelism

4.2 The Computational Effort

To estimate the amount of computational effort needed in a given section of the mesh, we employ some heuristics. Here we use the physics of the problem to help determine a good decomposition.

4.2.1 Mean Free Path Estimate

The first heuristic we examined, *mean free path* (mfp) estimation, uses the scattering mean free path ($\lambda = 1/\mu$) and the distance from the particle source (r) to estimate the work that will occur in a zone. Our problems used point sources of particles, so we estimate the work in zone i as

$$W_i = \frac{\mu}{r}. \tag{7}$$

4.2.2 Coarse Grid Estimate

Coarse grid estimation uses a lower resolution version of the given problem to estimate the work distribution. A smaller number of particles are tracked through a coarser version of the computational mesh, keeping track of the amount of effort involved in each zone. The work estimate for a zone in the fine mesh will then be the actual work for the coarse zone that contains it.

4.3 Decomposition

We decomposed the problems into two subdomains that had approximately equal numbers of zones. The work estimate is used for each subdomain to decide how many times we would replicate each subdomain on the processors. The decision to use two subdomains was arbitrary; other problems may require more subdomains due to memory constraints. A problem could be divided into more that two subdomains by recursively decomposing the subdomains.

Initially, each subdomain was assigned to one processor. We assigned the remaining processors one at a time, always assigning the next processor to the subdomain with the highest estimated work per processor. Provided the work estimate is accurate, this will provide the best load balance possible.

We used two different decomposition methods to divide the mesh into two subdomains. The first decomposition algorithm found an estimated work value w_c such that the sets $H = \{i | w_i > w_c\}$ and $L = \{i | w_i \leq w_c\}$ had approximately equal cardinality (i.e. w_c is close to the median work estimate for a zone). For the types of problem we have run, this simple decomposition method yields two contiguous domains. For this paper, we refer to this as *Median Work Estimate* (mwe) decomposition.

The second method cut the mesh along a plane parallel to a user-defined plane. The decomposition algorithm selected a plane parallel to one defined by the user that divided the mesh into two subdomains with equal amounts of estimated work. This method ignored the sizes of the subdomains. For the test problems in this paper, it generated similar-sized subdomains. A more sophisticated algorithm may be necessary for other problems. For this paper, we refer to this as *axis-wise* decomposition.

5 Results of Some Test Problems

5.1 The Test Problems

We used the algorithms discussed to decompose the mesh for three test problems. Each problem uses a 20x20x20 grid. The material properties vary for the different problems. The relevant material properties for the Monte Carlo module are the scattering coefficient μ_s and the absorption coefficient μ_a.

Uniform: In the first test problem, **uniform** (see Fig 4), the material properties are uniform over the entire mesh. The attenuation coefficient is typical for undamaged biological

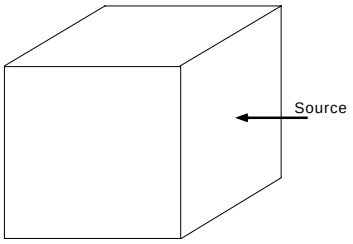


Figure 4: Uniform test problem. The material is homogeneous

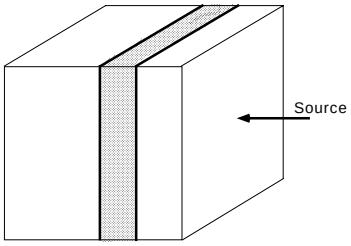


Figure 5: Near Wall test problem. The slab is optically thick.

tissue. The source is a collimated beam incident on one face of the mesh.

Near Wall: The **near wall** test problem (Fig. 5) also has a beam source. There is a slab of material with a higher attenuation coefficient normal to the source. The attenuation coefficient for the slab is high enough that few particles will penetrate to the material behind the slab.

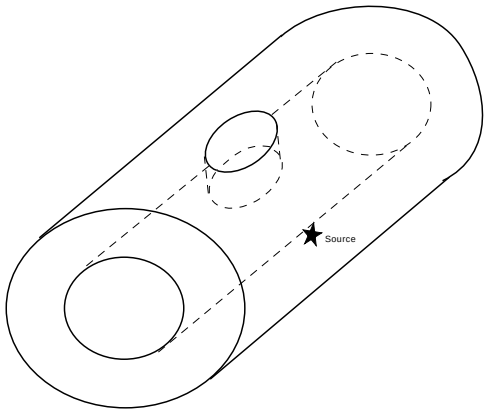


Figure 6: Tube Test problem. The tube is optically thick, the source is isotropic

Tube with a Hole: Test problem **tube** (See Fig. 6) is a true three dimensional problem. It has a tube of material that is optically thick running parallel to one axis of the mesh. The tube has a circular hole cut in one side. The particle source is isotropic and placed inside the tube, opposite the hole. This is the first part of an effort to simulate endovascular patch welding [12]—previously done in two dimensions—in three dimensions.

For each test problem, five separate series of runs were made. The first run used a conventional decomposition method typical of finite element problems. The mesh was divided into as many disjoint subdomains as there were processors, and the problem run. We refer to this as *geometric* decomposition in this paper.

In addition to the geometric decomposition, we used both work estimation methods (mean free path and coarse grid estimation) in combination with each method of dividing the computational mesh in two (median work estimate and axis-wise decomposition).

The geometric decomposition will tend to use less memory. It has as many subdomains as processors, where the hybrid method—as used in this paper—always uses two subdomains. The comparison of the geometric decomposition with the hybrid decomposition is useful to see the difference between a method that does not consider the locality of Monte Carlo transport (geometric) and one that does (the hybrid method).

5.2 Numerical Results and Discussion

The test problems were run on a Digital AlphaServer 8400 (the "DEC machine") at the Lawrence Livermore National Laboratory computing center. The DEC machine had eight processors available. The code used the MPI library to handle message passing.

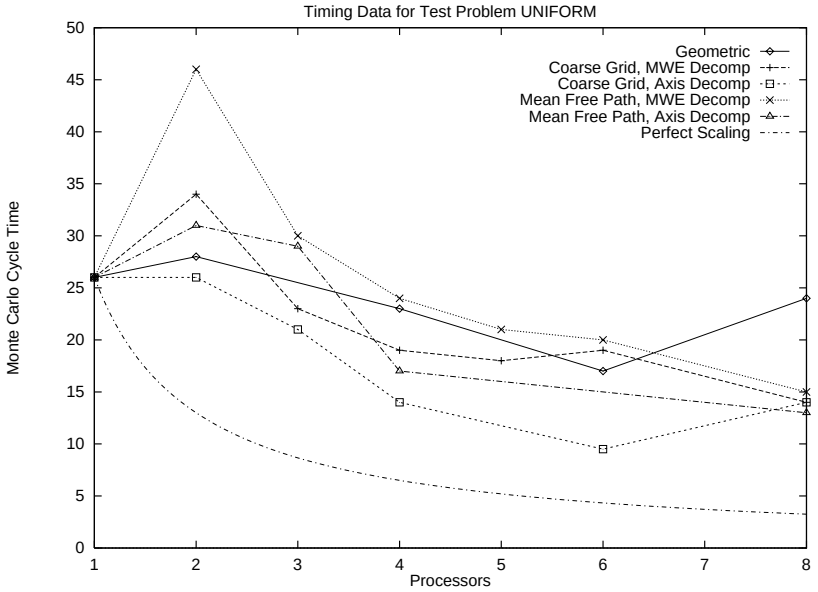


Figure 7: Timing Results for test problem **uniform**

The speedup results for the three test problems are plotted in Figs. 7, 8, and 9. For each problem, the total time for a single Monte Carlo cycle is plotted against the number of processors. Each plot has a separate line for each decomposition algorithm used. If the scaling were perfect, the time would decrease as t_1/N, where t_1 is the time on one processor and N is the number of processors. The t_1/N curve is on each plot, labeled "Perfect Scaling."

Timing results for the test problems **uniform** and **nearwall** (Figs. 7 and 8) are encouraging. In both cases, the hybrid decomposition approach outperformed the geometric decomposition. In test problem **nearwall**, the hybrid approach ran twice as fast. For these test problems, the hybrid decomposition scheme achieved better load balance and lowered communication costs.

The results for test problem **tube** (Fig. 9) were less impressive. The hybrid scheme performed as well as geometric decomposition in most cases. One of the mean free path decomposition schemes performed very poorly; this is due to the median work estimate algorithm generating a poor decomposition, where the subdomains were not contiguous.

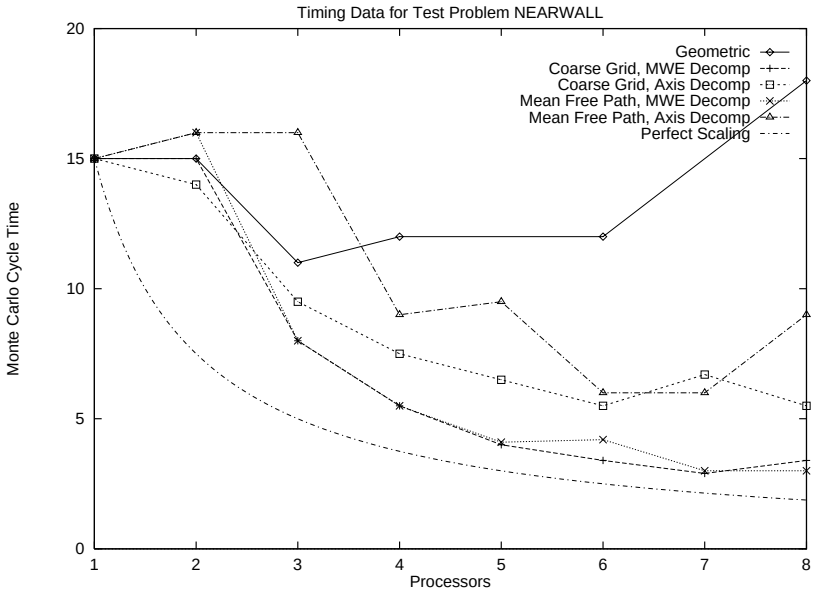


Figure 8: Timing Results for test problem **nearwall**

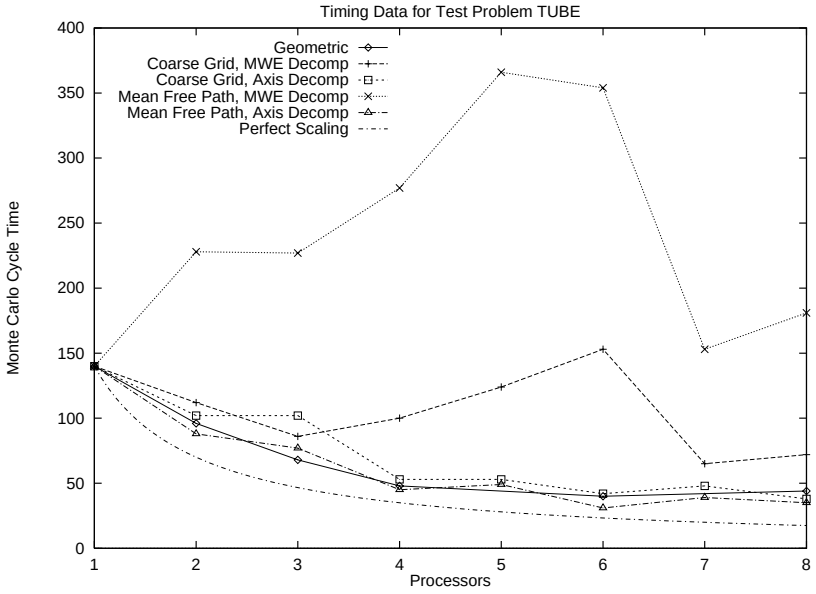


Figure 9: Timing Results for test problem **tube**

This greatly increased communication time compared to computation time for this problem, as particles crossed many more inter-processor boundaries.

Closer examination—in progress–of the results of the test problems indicate that, with the exception of the problems discussed above, load balance is the most important factor affecting performance of the parallel code. The time spent idle due to communication latency was dwarfed by the time spent idle waiting for other processors to complete work.

One source of load imbalance came from the communication scheme used in the code. A typical decomposition resulted in some subdomains being replicated more often than others. This resulted in an unbalanced communication scheme, which causes a load imbalance.

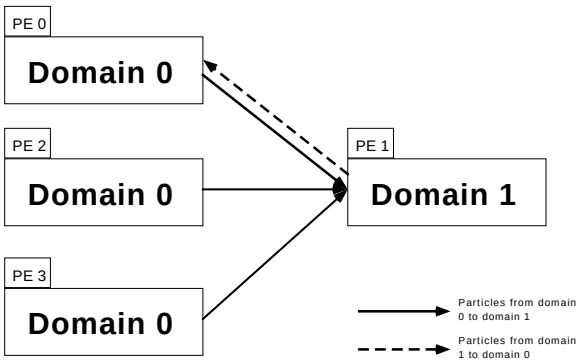


Figure 10: Unbalanced communication scheme can cause load imbalance

Fig. 10 illustrates the problem. Processor 1 has domain 1, it communicates with several processors with domain 0. It receives particles from all of the processors with domain 0, but only sends particles to processor 0. This tends to give processor 0 more work to do than processor 2 and processor 3, causing a load imbalance. This problem is being addressed by modifying the communication scheme so that a processor may split its outgoing particles for a domain among the processors that have that domain.

The axis-wise decompositions were most effective in achieving load balance for the problems tested. We saw the best speedups over geometric decomposition for test problem **nearwall**.

6 Summary and the Future

Effective use of parallel computers is important in facilitating the move to the three dimensional codes that are the future of computational physics. We have examined some methods for parallelizing Monte Carlo transport calculations that take into account the locality of Monte Carlo. The examples come from laser-tissue modeling, but the lessons learned can help in other types of transport. Domain decomposition without reference to the physics or the numerics of a calculation that uses Monte Carlo transport will be problematic.

The hybrid task farm-spatial scheme outlined in Sec. 4.1.3 shows great promise as a method for using parallelism for computational models that have a Monte Carlo transport

package working in concert with other packages. It allows all packages to benefit from parallelism, but provides the most enhancement to the Monte Carlo package, which typically consumes the largest part of the computation time.

In the future, we hope to improve the algorithms that decompose the mesh by attempting to minimize the amount of time that must be spent communicating between processors. For example, a modified version of the RSB algorithm mentioned in Sec. 3.1 could generate decompositions that account for the communication required across an interface.

Another option being considered is a dynamic decomposition algorithm. In a time dependent problem, the actual computational effort expended per zone can be tracked and the decomposition modified based on that information for future cycles.

7 Acknowledgment

The authors wish to thank X Division and the Institute for Scientific Computing Research (ISCR) at the Lawrence Livermore National Laboratory for their support in this research.

References

[1] Bruce Hendrickson, Robert Leland, *An Improved Spectral Graph Partitioning Algorithm for Mapping Parallel Computations*, Technical Report SAND92-1460, Sandia National Laboratories, 1992.

[2] Richard A. London, Michael E. Glinsky, George B. Zimmerman, David C. Eder, Steven L. Jacques, Coupled Light Transport-Heat Diffusion model for Laser Dosimetry with Dynamic Optical Properties, *SPIE Proceedings, Laser-Tissue Interaction VI*, San Jose, CA, Feb 1995, v. 2391, p435

[3] S. A. Prahl, M. Keijzer, S. L. Jacques, A. J. Welch, A Monte Carlo Method of Light Propagation in Tissue, *SPIE Institute Series* v. IS 5 (1989)

[4] Richard A. London, Michael E. Glinsky, George B. Zimmerman, David S. Bailey, Steven L. Jacques, Laser-tissue interaction modeling with LATIS, *Applied Optics*, v. 36 (1 Dec. 1994), pp.9068-74

[5] Chang-Ming Ma, Implementation of a Monte Carlo Code on a Parallel Computer System, *Parallel Computing*, v. 20 (1994), pp. 991-1005

[6] F. Schmidt, W. Dax, M. Luger, Experiences with the Parallelization of Monte Carlo Problems, *Progress in Nuclear Energy*, v. 24 (1990), pp. 141-51

[7] J. Wood, H. Al-Bahadili, S. A. Khaddaj, Monte Carlo Photon Transport in Parallel Computers, *Progress in Nuclear Energy*, v. 24 (1990), pp. 153-64

[8] J. Wood, H. Al-Bahadili, S. A. Khaddaj, A Comparison of Monte Carlo Photon Transport on Two Types of Parallel Computer, *Annals of Nuclear Energy*, v. 18 (1991), n. 3, pp. 155-66

[9] C. Zhao, J. Wood, The Monte Carlo Method on a Parallel Computer, *Annals of Nuclear Energy*, v. 16 (1989), n. 12, pp. 649-57

[10] Ora E. Percus and Malvin L. Kalos, Random Number Generators for MIMD Parallel Processors, *Journal of Parallel and Distributed* Computing v. 6 (1989), pp.477-97

[11] A. De Matteis, S. Pagnutti, Controlling correlations in parallel Monte Carlo, *Parallel Computing*, v. 21 (1995), pp.73-84

[12] Michael E. Glinsky, Richard A. London, George B. Zimmerman, Steven L. Jacques, Joseph D. Ols, Modeling of endovascular patch welding using temperature feedback, *Proceedings of Medical Applications of Lasers III*, Barcelona, Spain, September 1995